A Unified Deep Metric Representation for Mesh Saliency Detection and Non-rigid Shape Matching

Shanfeng Hu, Hubert P. H. Shum, *Senior Member, IEEE,*, Nauman Aslam *Member, IEEE,*, Frederick W. B. Li and Xiaohui Liang

***Abstract*—In this paper, we propose a deep metric for unifying the representation of mesh saliency detection and non-rigid shape matching. While saliency detection and shape matching are two closely related and fundamental tasks in shape analysis, previous methods approach them separately and independently, failing to exploit their mutually beneficial underlying relationship. In view of the existing gap between saliency and matching, we propose to solve them together using a unified metric representation of surface meshes. We show that saliency and matching can be rigorously derived from our representation as the principal eigenvector and the smoothed Laplacian eigenvectors respectively. Learning the representation jointly allows matching to improve the deformation-invariance of saliency while allowing saliency to improve the feature localization of matching. To parameterize the representation from a mesh, we also propose a deep recurrent neural network (RNN) for effectively integrating multi-scale shape features and a soft-thresholding operator for adaptively enhancing the sparsity of saliency. Results show that by jointly learning from a pair of saliency and matching datasets, matching improves the accuracy of detected salient regions on meshes, which is especially obvious for small-scale saliency datasets, such as those having one to two meshes. At the same time, saliency improves the accuracy of shape matchings among meshes with reduced matching errors on surfaces.**

***Index Terms*—mesh saliency, non-rigid shape matching, metric learning, deep learning, recurrent neural network**

I. Introduction

The fundamental challenge of shape analysis is extracting knowledge from surface meshes that is not only understandable to humans but also invariant to complex shape deformations. Only with such invariance can a method work well consistently on the deformed versions of shapes. In this paper, we narrow this challenge down to two fundamental shape analysis tasks: *mesh saliency detection* [1] and *non-rigid shape matching* [2]. We develop their previously unknown underlying relationship, and exploit it for mutual improvements of saliency detection and shape matching using deep learning.

The first task we are interested in is saliency detection, which aims to compute a saliency map for an input mesh that signifies the perceptual or semantic importance of surface regions [1], [3]. Despite highlighting semantically important regions, saliency maps are also found to be consistent on surfaces of the same object category [3]. However, the intra-category consistency of saliency has been ignored by previous saliency detection methods [1], [3]–[5], which limits their generalization abilities under complex intra-category shape deformations.

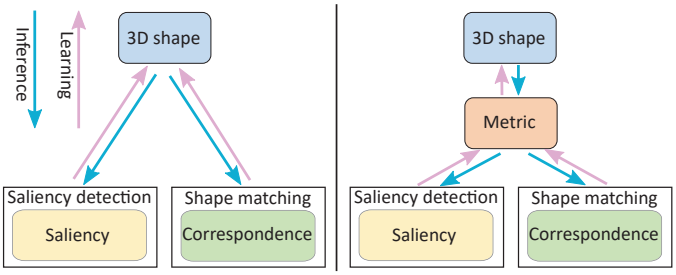


Fig. 1. **An Illustration of Our Idea**. While previous research approaches saliency detection and non-rigid shape matching separately and independently (*left*), we unify them via a shared metric representation of surface meshes to better handle intra-category shape deformations for both sides (*right*).

The other task we focus on is non-rigid shape matching, which finds semantically meaningful surface correspondences across meshes irrespective of the shape deformations among them [2]. As found in [3], human annotators tend to agree on a consistent set of semantically important regions on surfaces of the same object category, without communicating with each other during annotation. This shows that saliency is a strong deformation-invariant cue of shape matching within a category. However, existing shape matching methods mostly work on the matching task solely [6]–[8], without exploiting the saliency cue for more robust matching.

In computer vision and image processing, the consistency of salient objects within image collections has long been observed and exploited to drive effective image co-saliency detection [9] and matching [10]. However, the connection between saliency and matching has largely been ignored in computer graphics and shape analysis. We propose to unify the two tasks in the same framework so that they can help each other generalize better under complex intra-category shape deformations. To do this in a principled way, we need a unified representation of surface meshes that is geometry-aware, supports the joint modeling of saliency and matching, and most importantly enables the knowledge transfer between saliency and matching for mutual improvements.

Towards this goal, we propose a unified metric representation that measures the pairwise semantic distances among

S. Hu, H. P. H. Shum (the corresponding author), and N. Aslam are with the Department of Computer and Information Sciences at Northumbria University, Newcastle upon Tyne, NE1 8ST, UK, email: {shanfeng.hu, hubert.shum, nauman.aslam}@northumbria.ac.uk

F. W. B. Li is with the Department of Computer Science at Durham University, South Road, Durham, DH1 3LE, UK, email: frederick.li@durham.ac.uk

X. Liang is with the State Key Lab. of Virtual reality Technology and Systems at Beihang University, XueYuan Road NO. 37, Beijing, 100191, China, email: liang_xiaohui@buaa.edu.cn

all points on a mesh. Through two principled optimization problems, we show that the saliency map and the shape embeddings of a mesh can be derived from the principal eigenvector and the smoothed Laplacian eigenvectors of the metric respectively (Fig. 1). Our joint modeling allows matching to transfer deformation-invariance (i.e. intra-category consistency) to saliency, while allowing saliency to transfer sparsity (i.e. semantic feature localization) to matching for more robust correspondence solutions.

Having found a unified metric representation for saliency detection and shape matching, we need a way to compute the metric from the low-level geometry features of all points on an input mesh. More importantly, we wish the computation process to be differentiable so that it can be automatically learned from a given pair of saliency and matching datasets. Witnessing the success of deep neural networks for shape analysis [8], we propose a deep metric learning architecture that maps the low-level geometry features of all points on a mesh to a semantics-aware metric representation for saliency detection and shape matching. The core of our architecture is a multi-layer RNN that can be learned to effectively integrate small-to-large scale shape features for each point. The other essential component of our architecture is a soft-thresholding operator, which can be learned to produce a sparse metric from the pooling result of the metrics computed from the RNN features of each scale. Our architecture is able to more effectively exploit multi-scale shape information and the sparsity of saliency, producing higher performance on saliency detection than alternatives.

To learn the deep metric representation from a pair of saliency and matching datasets, we propose a unified loss function with three terms: (1) the saliency fitting term to penalize the difference between the predicted and the ground-truth saliency maps of a mesh from the saliency dataset; (2) the saliency consistency term to penalize the difference between the predicted saliency maps of any pair of meshes from the matching dataset; (3) the metric consistency term to penalize the difference between the two metrics of any pair of meshes from the matching dataset. We minimize this loss function using our proposed eigenvector reparameterization technique with the stochastic gradient descent (SGD) method [11].

We jointly evaluate our method on saliency detection [3] and non-rigid shape matching [12]–[14] datasets. The results show that it outperforms exiting rule-based and learning-based saliency detection methods in both the small and large sample training scenarios. It is also shown to improve both the model-based and learning-based methods for matching non-isometric pairs of shapes (Fig. 2). Our publicly available source code can be downloaded from this link: https://drive.google.com/drive/folders/10Vu3ujF-5gPm8h_E35VhZR45WCjht18B

Our contributions include:

- We validate the mutual benefits between mesh saliency detection and non-rigid shape matching. Matching improves the accuracy and deformation-invariance of saliency via the intra-category consistency of matching, while saliency improves the robustness of matching under non-isometric deformations via the sparsity of saliency.
- We propose a unified metric representation for joint modeling of saliency and matching. The saliency map of a mesh is computed as the principal eigenvector of the metric and the shape embeddings of the mesh are computed as the smoothed Laplacian eigenvectors of the metric. Our formulation allows matching to enforce the intra-category consistency for more accurate and deformation-invariant saliency detection, while exploiting the sparsity of saliency to induce semantically localized embeddings for more robust matching.
- We propose a multi-layer RNN architecture for more effectively integrating multi-scale shape information in metric computation, and an effective soft-thresholding operator for incorporating the sparsity of saliency in metric representation. We also propose a unified loss function for joint metric learning from a pair of saliency detection and shape matching datasets.

In the following, we review existing work in Section II and present our unified metric representation for saliency detection and shape matching in Section III. We then describe our deep metric learning architecture in Section IV and some implementation details in Section V. We present results in Section VI and draw our conclusions in Section VII.

II. Related Work

Mesh saliency was introduced to computer graphics to measure the perceptual or semantic importance of surface regions [1]. Traditional methods computed saliency either from local contrasts [1], [5] or global rarities [4], [15], [16]. These hand-crafted saliency rules are neither accurate nor robust to non-rigid shape deformations.

Recent methods directly learned a saliency prediction function from human annotations [3]. The 3D deep neural networks of [17]–[22] can also be adapted for saliency prediction. We follow the Schelling saliency notion of [3] in this work as they empirically validated the intra-category consistency of Schelling saliency maps. Still, we find no previous methods enforcing this property for saliency detection. In contrast, our method explicitly enforces the intra-category consistency of saliency and produces more accurate and deformation-invariant saliency maps.

Non-rigid shape matching finds semantically meaningful surface correspondences across meshes irrespective of the deformations among them [2]. Traditional methods mainly assumed the deformation to be isometric [6] or conformal [7] and then searched for matchings within the prescribed deformation space. Due to the isometry-invariant property, the surface Laplacian [23] has been widely used in the spectral embedding [24], functional mapping [6], and quadratic matching [25] formulations of shape matching. Both isometric and conformal deformations are restricted and can bias shape matching towards unfavorable solutions.

Recent methods learned deformation-invariant shape embeddings for correspondence search [26]–[30] or directly learned point label classifiers for correspondence prediction using random forests [31], convolutional neural networks (CNNs) [8], [32], and multi-layer perceptrons (MLPs) [33].

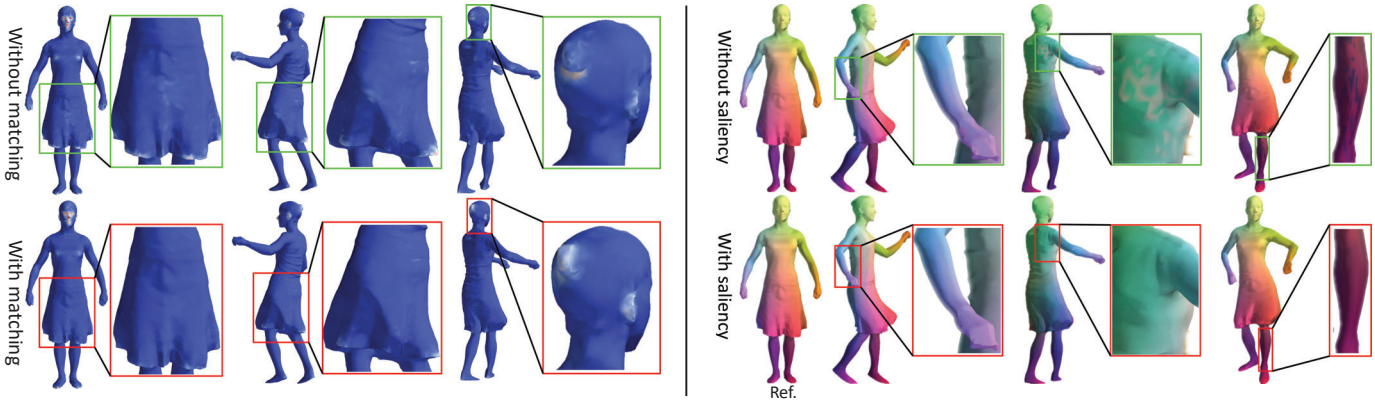


Fig. 2. **The Mutual Benefits of Saliency and Matching**. Our method produces more deformation-invariant saliency maps with matching (*left*, using red and blue colors to visualize high and low saliency values respectively). It also produces more accurate shape matchings with saliency (*right*, colorizing each target mesh vertex with its computed corresponding reference vertex's (X,Y,Z) coordinates).

Both streams of methods learned for individual points without considering their saliency information. Also, the first stream of methods learned embeddings on a per-point basis and lacked orthogonality and smoothness guaranties that hold for the Laplacian embeddings [23]. Our method instead guarantees that the learned embeddings are orthogonal (i.e. the inner product between every pair of embedding vectors is zero) and smooth. More importantly, it exploits saliency to ensure that the resulting embeddings are localized on semantically important surface regions. This is particularly valuable in improving both the model-based and learning-based methods for matching non-isometric pairs of shapes.

3D shape recognition can also be benefited by the joint use of mesh saliency and shape matching. Traditionally, shape recognition was generally performed by computing the similarity of geometric descriptors extracted from shapes [34]–[37], with some benchmarks specifically built to evaluate the effectiveness of these descriptors [38], [39]. Recently, deep learning methods have also been adopted for 3D shape recognition [18]–[20], [36], [40] using considerably larger-scale shape datasets [41], [42]. As mesh saliency is remarkably consistent within shapes of the same object class [3], our saliency-guided shape embeddings could also be summarized into shape descriptors that are sufficiently deformation-invariant to allow more robust shape recognition.

III. Our Unified Metric Representation

In this section, we propose a unified metric representation of surface meshes that enables the joint modeling of saliency detection and non-rigid shape matching. While multi-task learning is traditionally formulated as learning shared feature representations, it would be based on individual points on a surface and therefore lack a global geometry characterization of the whole surface [3], [18]–[20]. In contrast, we propose to represent the geometry of a mesh using a metric that characterizes the pairwise learned distances among all points on the surface. We will show that such a global metric representation is essential to guaranteeing some desirable properties for saliency detection and shape matching.

A. Notations, Inputs, and Outputs

We denote a polygonal surface mesh as $\mathcal{P} = \langle \{\boldsymbol{p}^k \in \mathbb{R}^3\}_{k=1}^N, \{(\boldsymbol{p}^i, \boldsymbol{p}^j) \mid \text{if } \boldsymbol{p}^i \text{ and } \boldsymbol{p}^j \text{ are adjacent}\} \rangle$ with N surface vertices and the edges connecting the adjacent vertices. One quantity we want to compute for $\mathcal{P}$ is a nonnegative-valued saliency map $\boldsymbol{s}(\mathcal{P}) \in \mathbb{R}_{\geq 0}^N$, which assigns to each point $\boldsymbol{p}^k$ the saliency value $s_k(\mathcal{P})$. The higher the value, the more semantically important the point. The other quantity we want to compute is the shape embeddings $\boldsymbol{E}(\mathcal{P}) \in \mathbb{R}^{N \times m}$, which maps each 3D point $\boldsymbol{p}^k$ to a m-dimensional feature vector $\boldsymbol{E}_{k\cdot}(\mathcal{P})$ where non-rigid shape deformations can be simplified to rigid ones for more efficient matching [25]. We denote the metric representation that leads to the two quantities as a nonnegative-valued, symmetric, and zero-diagonal distance matrix $\boldsymbol{D}(\mathcal{P}) \in \mathbb{R}_{\geq 0}^{N \times N}$. It assigns a distance $\boldsymbol{D}_{ij}(\mathcal{P})$ to every pair of points $\boldsymbol{p}^i$ and $\boldsymbol{p}^j$ on the surface of $\mathcal{P}$.

In order to learn the metric $\boldsymbol{D}(\mathcal{P})$ for saliency detection and non-rigid shape matching, we require a pair of saliency and matching datasets, $\{\langle \mathcal{P}_i, \bar{\boldsymbol{s}}(\mathcal{P}_i) \rangle\}_{i=1}^{N_s}$ and $\{\langle \mathcal{P}_i, \mathcal{P}_i' \rangle\}_{i=1}^{N_c}$, for training. In the former, each mesh $\mathcal{P}_i$ has the ground-truth saliency map $\bar{\boldsymbol{s}}(\mathcal{P}_i)$. In the latter, every pair of meshes $\mathcal{P}_i$ and $\mathcal{P}_i'$ have a natural one-to-one semantic correspondence between their surface points.

B. Saliency Detection from a Metric

In this subsection, we propose a differentiable saliency definition based on the metric representation of a mesh. We formulate the saliency map of a mesh as the global optimal solution to a metric-based optimization problem, obtaining the solution as the principal eigenvector of the metric. This solution is differentiable and thus learnable, guarantees the nonnegativity of saliency, and inherently encodes the sparsity of saliency for saliency detection and shape matching.

To begin with, we first consider $\boldsymbol{s}(\mathcal{P})$ as a binary saliency map: $s_k(\mathcal{P}) = 1$ if $\boldsymbol{p}^k$ is a salient point and $s_k(\mathcal{P}) = 0$ otherwise. We then consider the problem of labeling a set of salient points so that the sum of their pairwise distances, $\boldsymbol{s}^T \boldsymbol{D}(\mathcal{P}) \boldsymbol{s} = \sum_i \sum_j s_i s_j \boldsymbol{D}_{ij}(\mathcal{P})$, can be maximized. Finally, since solving this problem is difficult and only produces a

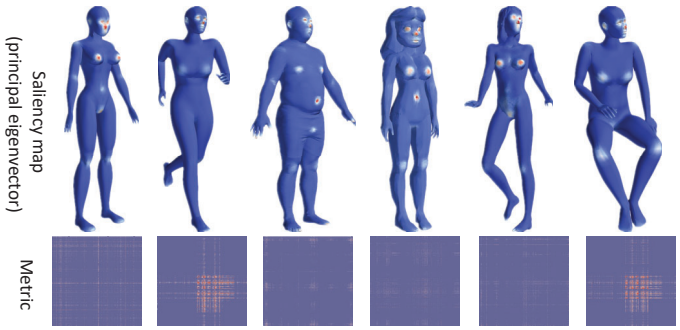


Fig. 3. **The Sparsity of Saliency**. As human-annotated saliency maps only highlight a few semantically important regions on surfaces [3], our system automatically learns to produce sparse metrics whose principal eigenvectors (i.e. computed saliency maps) are sparse as well. Here, a redder matrix element represents a larger learned distance between the corresponding pair of surface points it visualizes.

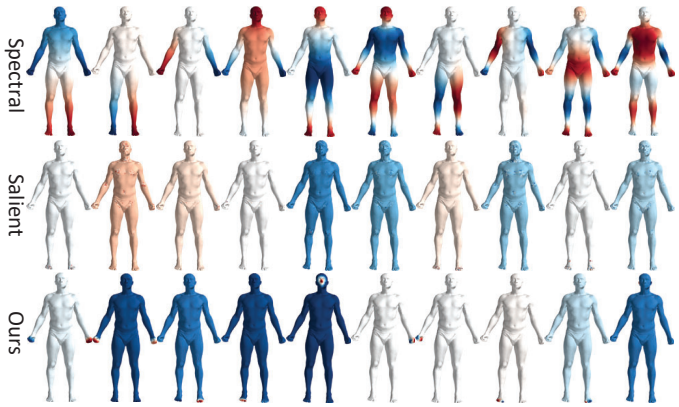


Fig. 4. **Our Saliency-induced Embeddings**. The columns from left to right show individual embedding components computed by three different methods, with the color visualizing the smoothness and localization of the embeddings on the surface. On top of being as locally smooth as the Laplacian spectral embeddings, our embeddings are further globally localized on semantically important surface regions (i.e. eyes, ears, and limbs). Therefore, they are able to enforce additional constraints for robust shape matching.

binary saliency map, we relax it as follows by replacing the binary saliency labels with continuous saliency values:

$$\boldsymbol{s}(\mathcal{P}) = \arg\max \boldsymbol{s}^T \boldsymbol{D}(\mathcal{P})\boldsymbol{s}, \text{s.t. } \boldsymbol{s} \in \mathbb{R}^N_{\geq 0} \text{ and } \|\boldsymbol{s}\|_2 = 1, \quad (1)$$

where we enforce the unit Euclidean norm* constraint for solution well-posedness. Without the nonnegativity constraint, the objective of the problem is known as the Rayleigh quotient of the metric $\boldsymbol{D}(\mathcal{P})$ and the solution that globally maximizes it is the principal eigenvector of $\boldsymbol{D}(\mathcal{P})$. Since the metric is symmetric and nonnegative-valued by definition, its principal eigenvector is unique and guaranteed to be nonnegative-valued according to the Perron-Frobenius theorem [43], [44]. Therefore, the optimal saliency map $\boldsymbol{s}(\mathcal{P})$ is the principal eigenvector of the metric $\boldsymbol{D}(\mathcal{P})$.

Compared to existing saliency detection methods of [1], [3]–[5], [15]–[20], our metric-based saliency detection method has the following desirable properties:

- **Nonnegative-Valued**. This is trivial but is not automatically satisfied by existing learning-based saliency detection methods, without the use of some non-linear activation functions that squash regression outcomes to nonnegative saliency values. In contrast, our saliency map $\boldsymbol{s}(\mathcal{P})$ is nonnegative-valued by definition.
- **Differentiable**. As the metric $\boldsymbol{D}(\mathcal{P})$ is symmetric, $\boldsymbol{s}(\mathcal{P})$ being one of its eigenvectors is continuously differentiable with respect to it [43]. This allows us to fit $\boldsymbol{s}(\mathcal{P})$ to the ground-truth saliency map $\bar{\boldsymbol{s}}(\mathcal{P})$ and the map of any corresponding mesh $\mathcal{P}'$, producing more accurate and deformation-invariant saliency maps than existing rule- and learning-based methods.
- **Encoding Sparsity**. Apart from the intra-category consistency, the other characteristic of the Schelling saliency maps is that they are sparse [3]. When fitted to them, our saliency map $\boldsymbol{s}(\mathcal{P})$ becomes sparse as well and drives a large fraction of the entries of the metric $\boldsymbol{D}(\mathcal{P})$ to zeros, which encode distances among non-salient points (Fig. 3). This sparsification mechanism is key to deriving semantically localized shape embeddings for more robust shape matching (III-C).

*$\|\boldsymbol{x}\|_2 = \sqrt{\sum_i \boldsymbol{x}_i^2}$

C. Non-rigid Shape Matching from a Metric

Having formulated the saliency map as the principal eigenvector of a metric in Section III-B, we now describe how to obtain a shape embedding matrix $\boldsymbol{E}(\mathcal{P})$ from the same metric for robust shape matching. Our idea is to exploit the sparsity of saliency to learn better discrimination for salient points and more invariance for non-salient points. To do this, we formulate $\boldsymbol{E}(\mathcal{P})$ as the Laplacian embeddings with the metric $\boldsymbol{D}(\mathcal{P})$ and the surface connectivity of a mesh, so that they can be smooth, orthogonal, semantically localized, and deformation-invariant.

The deformation between two real-world shapes is generally non-rigid, making it highly challenging to be handled in the original 3D Euclidean space. Following the framework of [25], we compute a set of discriminative and deformation-invariant embedding coordinates $\boldsymbol{E}_{k\cdot}(\mathcal{P})$ for each surface point $\boldsymbol{p}^k$, so that the non-rigid shape deformation between a pair of meshes $\mathcal{P}$ and $\mathcal{P}'$ in the original 3D Euclidean space can be simplified into an approximately rigid one in the higher-dimensional embedding space. While existing methods strive on the discrimination and invariance of shape embeddings [26]–[30], they learn for each individual surface point separately and therefore cannot guarantee that the obtained embeddings are orthogonal or smooth. Moreover, they treat all points equally and ignore the fact that salient points are semantically more important and geometrically more consistent within a shape category [3].

To address these issues, we consider the following Laplacian

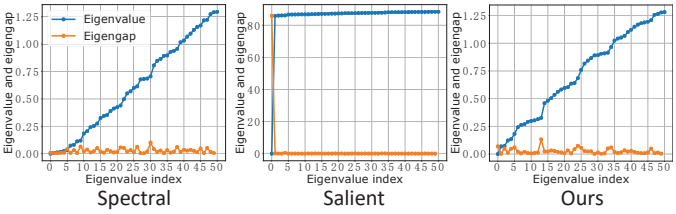


Fig. 5. **The Deformation Stability of Our Embeddings**. Both the Laplacian spectral and our saliency-induced embeddings have non-zero eigengaps. Therefore, they can be made stable under complex intra-category shape deformations if the two metrics of any pair of meshes can be learned to be consistent within a shape category.

embedding problem [45]:

$$\boldsymbol{E}(\mathcal{P}) = \arg\min \mathrm{tr}[\boldsymbol{E}^T \Delta[\boldsymbol{A}(\mathcal{P})]\boldsymbol{E}], \tag{2a}$$

$$= \arg\min \underbrace{\frac{1}{2}\sum_{k=1}^{m}\sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{A}_{ij}(\mathcal{P})(\boldsymbol{E}_{ik}-\boldsymbol{E}_{jk})^2}_{\text{affinity-weighted smoothness penalties}}, \tag{2b}$$

$$\text{subject to } \underbrace{\boldsymbol{E} \perp \mathbf{1} \text{ and } \boldsymbol{E}^T\boldsymbol{E} = \boldsymbol{I}}_{\text{orthogonality constraints}}, \tag{2c}$$

where $\mathrm{tr}[\cdot]$ is the matrix trace[†], $\Delta[\cdot]$ is the graph Laplacian, and $\boldsymbol{A}(\mathcal{P}) = (1-\theta)\boldsymbol{C}(\mathcal{P}) + \theta\boldsymbol{S}(\mathcal{P})$ is a convex combination of the cotangent affinity matrix $\boldsymbol{C}(\mathcal{P})$ [23] and the salient affinity matrix $\boldsymbol{S}(\mathcal{P})$ of a mesh. We compute the salient affinity matrix $\boldsymbol{S}(\mathcal{P})$ by first computing $1 - \boldsymbol{D}(\mathcal{P})$ and then setting the diagonal elements of the result to zeros. That is, $\boldsymbol{S}_{ij}(\mathcal{P}) = 1 - \boldsymbol{D}_{ij}(\mathcal{P})$ if $i \neq j$, and 0 otherwise. While $\boldsymbol{C}(\mathcal{P})$ captures the affinities of adjacent surface points and $\boldsymbol{S}(\mathcal{P})$ encodes considerably large affinities among non-salient points, $\boldsymbol{A}(\mathcal{P})$ is a balance of them. As $\Delta[\boldsymbol{A}(\mathcal{P})]$ is symmetric and nonnegative-definite, it is known that the optimal embeddings $\boldsymbol{E}(\mathcal{P})$ are its eigenvectors associated with the $m+1$ smallest eigenvalues (excluding the constant eigenvector corresponding to the eigenvalue of 0) [45].

Compared with the shape embeddings of [26]–[30], our saliency-induced ones have the following desirable properties:

- **Orthogonal**. Because $\Delta[\boldsymbol{A}(\mathcal{P})]$ is symmetric, the embedding coordinates $\boldsymbol{E}(\mathcal{P})$ being m of its eigenvectors are orthogonal to each other by definition. Therefore, our shape embeddings are mutually uncorrelated as the Laplacian spectral embeddings [23].
- **Smooth**. When setting θ to 0, we recover the Laplacian spectral embeddings of a mesh from (2a,2b,2c), which are the smoothest orthogonal functions on the surface [23] (Fig. 4, top). By setting θ to 0.1 to account for the affinities of adjacent surface points, we are able to ensure that our embeddings are smooth and orthogonal at the same time (Fig. 4, bottom).
- **Semantically Localized**. When setting θ to 1, we obtain embeddings that are localized on salient points (Fig. 4, middle), as the embedding smoothness among non-salient points is heavily enforced due to their much larger learned mutual affinities. Empirically, by setting θ to 0.1, we obtain both smooth and semantically localized embeddings (Fig. 4, bottom). Setting θ to a larger or a smaller value would weaken the smoothness or the localization property.
- **Deformation-Invariant**. According to the Davis-Kahan theorem described in [46], we have the following bound on the distance between the shape embeddings of two meshes:

$$\mathrm{d}(\boldsymbol{E}(\mathcal{P}), \boldsymbol{E}(\mathcal{P}')) \leq \frac{\|\Delta[\boldsymbol{A}(\mathcal{P})] - \Delta[\boldsymbol{A}(\mathcal{P}')]\|_F}{\lambda_{m+1} - \lambda_m}, \tag{3}$$

where $\mathrm{d}(\cdot,\cdot)$ is the Euclidean norm of the sines of the principal angles between $\boldsymbol{E}(\mathcal{P})$ and $\boldsymbol{E}(\mathcal{P}')$, and $0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$ are the non-decreasing eigenvalues of $\Delta[\boldsymbol{A}(\mathcal{P})]$. To lower this bound, we need to decrease its numerator by enforcing the deformation-invariance of the pair of learned metrics $\boldsymbol{D}(\mathcal{P})$ and $\boldsymbol{D}(\mathcal{P}')$. We also set $\theta = 0.1$ to ensure its denominator (the eigengap) is non-negligible, preventing divergence of the bound (Fig. 5). This ensures that our embeddings are sufficiently deformation-invariant for shape matching.

[†]$\mathrm{tr}[\boldsymbol{X}] = \sum_i \boldsymbol{X}_{ii}$, $\Delta[\boldsymbol{X}]_{ii} = \sum_j \boldsymbol{X}_{ij}$ and $\Delta[\boldsymbol{X}]_{ij} = -\boldsymbol{X}_{ij}$ if $i \neq j$

IV. Our Deep Metric Learning Architecture

In Section III, we have proposed a unified metric representation of surface meshes whose principal eigenvector and smoothed Laplacian eigenvectors can be used for saliency detection and non-rigid shape matching respectively. In this section, we propose a deep neural network architecture for computing the metric from an input mesh. The reason we need a deep architecture is that it is learnable and sufficiently powerful to extract high-level features from low-level geometry data for shape analysis [8], [18], [19], [33]. As shown in Fig. 6, for each point on a surface mesh, we first (a) extract a set of raw multi-scale feature vectors and then (b) feed them into our proposed multi-layer RNN for multi-scale feature embedding. We then (c) compute a set of multi-scale Euclidean metrics to (d) derive a scale-free metric via max-pooling. Afterwards, we (e) use our proposed soft-thresholding operator to adaptively sparsify this metric and (f) compute the principal eigenvector to (g) form three loss terms. Finally, we minimize these terms together using our proposed eigenvector reparameterization trick with the SGD method [11].

A. Our RNN for Multi-scale Feature Embedding

In this section, we describe our RNN method for multi-scale feature embedding. The inputs to our method are the raw multi-scale shape descriptors of a mesh $\mathcal{P}$, $\{\boldsymbol{F}^{\tau,0}(\mathcal{P}) \in \mathbb{R}^{N\times d}\}_{\tau=1}^{N_\tau}$, where N_τ is the number of scales from small to large and d is the feature dimension of each surface point at each scale τ. The outputs produced by our method are the embedded multi-scale features $\{\boldsymbol{F}^{\tau}(\mathcal{P}) \in \mathbb{R}^{N\times d}\}_{\tau=1}^{N_\tau}$, which are used for subsequence metric computations. As the shape information of each surface point naturally spans increasingly larger contexts and these contexts are not independent of each other [24], [47], it is difficult for some hand-crafted rules to discover the optimal correlation among multiple contexts and integrate them effectively [20]. This motivates us to consider

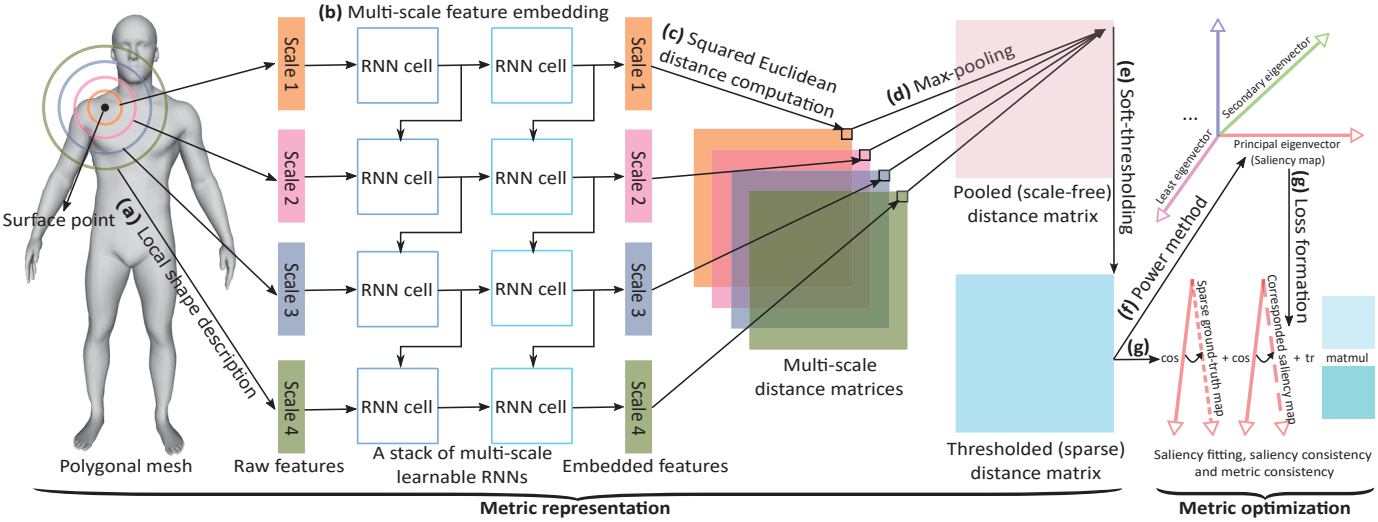


Fig. 6. **The Overview of Our Deep Metric Learning Architecture**. Steps (a)-(e) are used to compute a metric from the raw multi-scale features of a mesh, whilst Steps (f)-(g) are used to form the saliency fitting loss, saliency consistency loss, and metric consistency loss for metric learning. As our method uses a metric for joint modeling of saliency detection (principal eigenvector) and shape matching (Laplacian embeddings), it naturally incorporates the structure of all surface points for inference and learning.

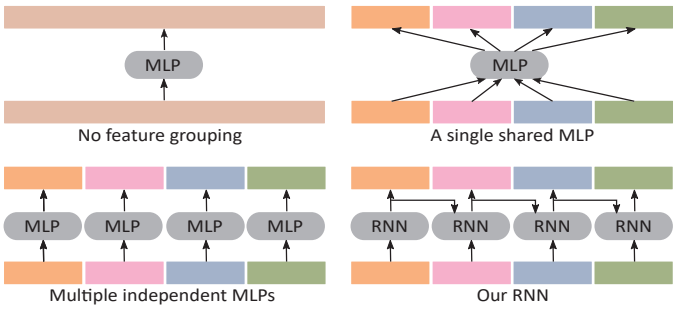


Fig. 7. **Multi-scale Feature Embedding Architectures**. The image shows three baselines and our RNN method for multi-scale feature embedding. A single MLP can transform the concatenated features of all scales jointly (*top left*) or the features of each scale individually (*top right*), and multiple MLPs can work on each scale separately with no feature sharing among each other (*bottom left*). In contrast, our RNN method works on a sequence of small-to-large scale features and explicitly learns the transition between scales for more effective scale integration (*bottom right*).

the multi-step RNN architecture that is usually popular for temporal sequence modeling.

Our idea is to learn features in two directions: the vertical direction that maps features from one layer to the next and the horizontal direction that propagates features from one scale to the next. More specifically, we propose to order shape features from small to large scales and then treat each scale as one step of a RNN in the scale sequence (Fig. 7, bottom right). This allows us to parameterize our feature embedding architecture as a multi-layer function $\boldsymbol{f} = \boldsymbol{f}^{N_l} \circ \cdots \circ \boldsymbol{f}^1$, each layer of which is a RNN with our specially designed scale interpolation cell structure as follows:

$$\boldsymbol{O}^{\tau,l} = \underbrace{\tanh[\boldsymbol{\Upsilon}(\boldsymbol{F}^{\tau-1,l}\boldsymbol{W}^{\diamond,l} + \boldsymbol{F}^{\tau,l-1}\boldsymbol{M}^{\diamond,l})]}_{\text{predicting candidate output features}}, \tag{4a}$$

$$\boldsymbol{P}^{\tau,l} = \underbrace{\text{sigmoid}[\boldsymbol{\Upsilon}(\boldsymbol{F}^{\tau-1,l}\boldsymbol{W}^{\circ,l} + \boldsymbol{F}^{\tau,l-1}\boldsymbol{M}^{\circ,l})]}_{\text{predicting scale interpolation weights}}, \tag{4b}$$

$$\boldsymbol{F}^{\tau,l} = \underbrace{\boldsymbol{\Upsilon}[(1-\boldsymbol{P}^{\tau,l}) \odot \boldsymbol{F}^{\tau-1,l} + \boldsymbol{P}^{\tau,l} \odot \boldsymbol{\Upsilon}(\boldsymbol{O}^{\tau,l})]}_{\text{interpolating features via convex combination}}, \tag{4c}$$

where l is the layer of each RNN, $\{\boldsymbol{W}^{\diamond,l}, \boldsymbol{M}^{\diamond,l}\}_{l=1}^{N_l}$ are the learnable matrix parameters of the RNN, and $\boldsymbol{\Upsilon}(\cdot)$ is the feature-wise standardization operator [48]. As a common practice [49], we initialize the features to zeros before the first step (i.e. scale 1) for RNN computation. To our knowledge, this is the first time that RNNs are used for multi-scale feature learning in shape analysis.

Compared with the alternatives shown in Fig. 7, our RNN learns scale integration explicitly to yield more powerful multi-scale features for shape analysis. Different from the vanilla RNN cell structure that only produces the features at each step [49], our cell in (4a,4c,4b) explicitly interpolates the features from the previous and the current steps (i.e. scales). Compared with long short-term memory (LSTM) [50] and gated recurrent unit (GRU) [51], our cell has a simpler and more effective scale integration mechanism for multi-scale feature embedding.

B. Our Soft-thresholding Operator for Metric Sparsification

Human-annotated saliency maps only highlight a few semantically important regions on surfaces [3]. However, existing saliency detection methods of [1], [3], [4], [15], [16] do not enforce the sparsity of saliency, producing excessive amounts of regions that are actually not salient (Fig. 14). This motivates us to directly incorporate the sparsity of saliency into metric representation for more accurate saliency detection (Fig. 8).

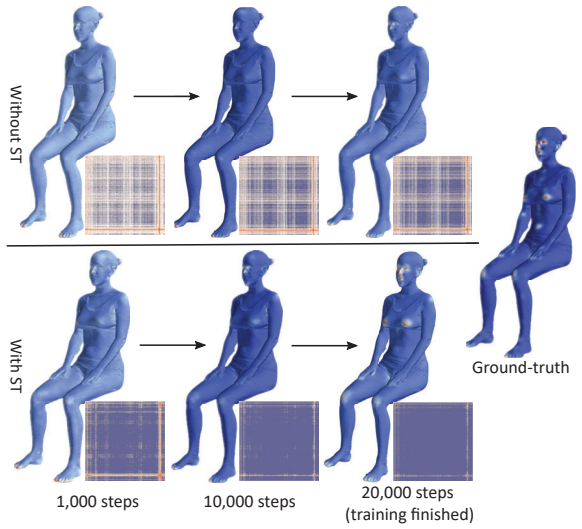


Fig. 8. **The Effect of Our Soft-thresholding Operator**. We learn a soft-thresholding (ST) operator to adaptively truncate the small elements of a metric to exact zeros, improving the sparsity and accuracy of computed saliency maps significantly.

Our idea is to adaptively soft-threshold a metric using a parametric threshold learned from ground-truth saliency maps. We propose our soft-thresholding operator as follows:

$$\boldsymbol{D}(\mathcal{P}) = \max\{\dot{\boldsymbol{D}}(\mathcal{P}) - \Theta_t, 0\}, \tag{5}$$

where $\dot{\boldsymbol{D}}(\mathcal{P})$ is the scale-free metric computed via max-pooling described below, and Θ_t is a scalar parameter that can be learned to truncate the small elements of $\dot{\boldsymbol{D}}(\mathcal{P})$ to exact zeros (see Section IV-C for analysis). This way, we can learn to sparsify $\dot{\boldsymbol{D}}(\mathcal{P})$ based on ground-truth saliency maps and ensure that its derived saliency map $\boldsymbol{s}(\mathcal{P})$ is properly sparsified as well.

We now describe how to compute $\dot{\boldsymbol{D}}(\mathcal{P})$. From the embedded features $\boldsymbol{F}^\tau(\mathcal{P})$ of each scale τ, we can compute a squared Euclidean distance matrix $\ddot{\boldsymbol{D}}^\tau(\mathcal{P})$ among the N points of a mesh [52]. We choose this representation because it is simple, differentiable, and analytically computable via fast matrix and vector operations. To address the rank-deficiency of a single Euclidean metric [52], we compute a scale-free metric by max-pooling the Euclidean metrics of all scales, $\dot{\boldsymbol{D}}(\mathcal{P}) = \max\{\ddot{\boldsymbol{D}}^1(\mathcal{P}), \ddot{\boldsymbol{D}}^2(\mathcal{P}), \ldots, \ddot{\boldsymbol{D}}^{N_\tau}(\mathcal{P})\}$, where the output is no longer low-rank as the linear independence of its rows (or columns) is greatly strengthened by the non-linear element-wise pooling operation.

Compared with traditional methods that enforce sparsity via a sparsity-inducing norm [53], ours learns sparsity by optimizing Θ_t adaptively, without the need of weighting a sparsity-inducing norm by trial-and-error. This leads to much more accurate and sparser saliency maps (Fig. 8).

C. Our Multi-objective Loss Function for Metric Learning

Here, we propose a loss function for metric learning from a given pair of saliency and matching datasets:

$$\mathcal{L}(\mathcal{P}, \mathcal{P}') = \alpha\mathcal{L}_\alpha(\mathcal{P}) + \beta\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}') + \gamma\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}'), \tag{6a}$$

$$\mathcal{L}_\alpha(\mathcal{P}) = 1 - \boldsymbol{s}(\mathcal{P})^T \bar{\boldsymbol{s}}(\mathcal{P}), \tag{6b}$$

$$\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}') = 1 - \boldsymbol{s}(\mathcal{P})^T \boldsymbol{s}(\mathcal{P}'), \tag{6c}$$

$$\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}') = 1 - \mathrm{tr}[\boldsymbol{D}(\mathcal{P})\boldsymbol{D}(\mathcal{P}')], \tag{6d}$$

where the *saliency fitting term* $\mathcal{L}_\alpha(\mathcal{P})$ penalizes the difference between the predicted and ground-truth saliency maps of a mesh from the saliency dataset, the *saliency consistency term* $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ penalizes the difference between the predicted saliency maps of any pair of meshes from the matching dataset, and the *metric consistency term* $\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}')$ penalizes the difference between the two metrics computed from any pair of meshes from the matching datasets. α, β, and γ are their respective weights.

Our Eigenvector Reparameterization. As the derivatives of $\boldsymbol{s}(\mathcal{P})$ with respect to $\boldsymbol{D}(\mathcal{P})$ require matrix pseudo-inverse [54], $\mathcal{L}_\alpha(\mathcal{P})$ and $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ cannot be minimized directly. We tackle this by approximating $\boldsymbol{s}(\mathcal{P})$ as follows:

$$\boldsymbol{s}(\mathcal{P}) \approx \frac{\boldsymbol{D}(\mathcal{P})\tilde{\boldsymbol{\nu}}}{\tilde{\boldsymbol{\nu}}^T \boldsymbol{D}(\mathcal{P})\tilde{\boldsymbol{\nu}}}, \tag{7}$$

where $\tilde{\boldsymbol{\nu}}$ is a numerical version of $\boldsymbol{s}(\mathcal{P})$ computed by the power iteration method. This approximation holds because $\tilde{\boldsymbol{\nu}}$ is associated with the largest eigenvalue of $\boldsymbol{D}(\mathcal{P})$ and is thus orthogonal to the other eigenvectors. Compared with the low-order approximation of [55], ours has a much simpler form and is significantly more accurate. In addition, our approximation is computationally efficient because the principal eigenvector of the nonnegative distance matrix $\boldsymbol{D}(\mathcal{P})$ is associated with the dominant principal eigenvalue [43], which guarantees that the ratio between the second largest absolute eigenvalue and the principal eigenvalue is strictly smaller than 1. As a result, the power iteration method to compute the principal eigenvector converges very quickly at a geometric rate. In practice, convergence normally takes fewer than 20 iterations of simple matrix and vector multiplication.

Our Saliency Fitting Term. To analyze learning dynamics, we insert (7) into (6b) to obtain the partial derivatives of $\mathcal{L}_\alpha(\mathcal{P})$ with respect to $\boldsymbol{D}(\mathcal{P})$ and Θ_t:

$$\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \boldsymbol{D}_{ij}(\mathcal{P})} = C_1 \tilde{\boldsymbol{\nu}}_i \tilde{\boldsymbol{\nu}}_j - C_2 \bar{\boldsymbol{s}}_i(\mathcal{P}) \tilde{\boldsymbol{\nu}}_j, \tag{8a}$$

$$\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \Theta_{\boldsymbol{t}}} = \sum_{i=1}^{N}\sum_{j=1}^{N} I\{\boldsymbol{D}_{ij}(\mathcal{P}) > \Theta_{\boldsymbol{t}}\} \frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \boldsymbol{D}_{ij}(\mathcal{P})}, \tag{8b}$$

where $0 < C_1 \leq C_2$ and $I\{\cdot\}$ is the indicator function. If the system wrongly predicts a low saliency value for $\boldsymbol{p}^i$, i.e. $\tilde{\boldsymbol{\nu}}_i < \bar{\boldsymbol{s}}_i(\mathcal{P})$, it will increase the distances of $\boldsymbol{p}^i$ to the other points because the derivatives $\{\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \boldsymbol{D}_{ij}(\mathcal{P})}\}_{j=1}^N$ are negative. Conversely, its distances to the other points will decrease. Sparse ground-truth saliency maps therefore leads to the sparsification of $\boldsymbol{D}(\mathcal{P})$. Because $\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \Theta_{\boldsymbol{t}}}$ is the sum of mostly negative partial derivatives from pairs of salient points whose distances are large enough to exceed the threshold,

Θ_t increases during the training to drive the sparsification of $\boldsymbol{D}(\mathcal{P})$ further (Fig. 4 and 8).

Our Saliency Consistency Term. The form of $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ is the same as that of $\mathcal{L}_\alpha(\mathcal{P})$, except that we treat $\boldsymbol{s}(\mathcal{P})$ and $\boldsymbol{s}(\mathcal{P}')$ as each other's learning target. This allows us to enforce the intra-category consistency of saliency by pushing the predicted saliency maps of any pair of meshes closer to each other in a shape category.

Our Metric Consistency Term. To obtain deformation-invariant embeddings, we need to control the bound in (3) by minimizing $\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}')$. This ensures that the learned metrics are sufficiently deformation-invariant. As the saliency consistency term $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ can only regularize the principal eigenvector of the learned metric, we add the metric consistency term $\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}')$ to control the remaining eigenvectors.

V. Implementation Details

We implement our proposed system in TensorFlow (V0.12). Our publicly available source code can be downloaded from this link: https://drive.google.com/drive/folders/10Vu3ujF-5gPm8h_E35VhZR45WCjht18B

Throughout our experiments, we stack three layers of RNNs with an input and an output dimension of 256 each. We initialize the matrix parameters of each RNN to be orthogonal, and we initialize the soft-thresholding parameter to zero. We set the learning rates for the RNN and the threshold parameters to 0.1 and 1×10^{-4} respectively, and decay them by a rate of 0.1 every 5,000 steps with a momentum of 0.9 for 20,000 SGD steps. We set the batch size to 1. We find that these hyper-parameters work well in our experiments, with the initial learning rates of the RNN and the threshold parameters being the most influential on the system performance. When a larger learning rate for either set of parameters is used, the training process does not converge well, degrading the final performance.

At each training step, we randomly retrieve a mesh and its ground-truth saliency map from a saliency dataset, as well as a pair of meshes from a shape matching dataset. We resample each mesh to 500 surface points for efficient learning and use all surface points for testing. We have experimented with varying numbers of sample points (including 500, 1,000, 1,500, and 2,000) and found that the performance of our system remains consistent within this range. A number smaller than 500 does not work well because the global shape features of surface meshes cannot be adequately captured by such sparse points. A number larger than 2,000 also tends to decrease the performance, since it essentially reduces the diversity of the samples from each mesh and, as a result, the size of the training set.

To learn a metric from a mesh, we use its spatial rather than spectral raw features, as the former capture both intrinsic and extrinsic geometry for shape analysis [56]. Specifically, we use the spherical harmonic (SH) descriptors of [47], which are derived from a raw distance field and have a theoretical guarantee of minimal information loss. We encode the local shape of each vertex with 16 SH amplitudes for each of 16 concentric shells of equally increasing radii, with the radius of the outmost shell being one-eighth of the mesh diameter. We pad the raw features of each scale with zeros to create a dimension of 256. This zero-padding does not impact the performance of our system because it is consistently applied to all scales in our experiments without introducing new information.

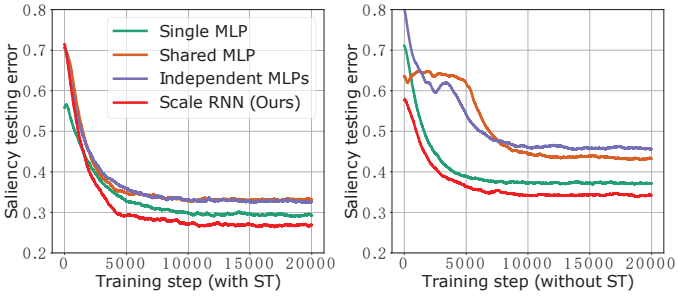


Fig. 9. **RNN Evaluations**. Our RNN method of learning and integrating multi-scale shape features produces the lowest saliency testing error, among the four feature embedding architectures in Fig. 7, both with (*left*) and without the soft-thresholding operator (*right*).

VI. Results

We train and test our system on a PC with an Intel Core i7-6700k CPU, 16GB RAM, and one Nvidia GTX 1080 graphics card with 8GB memory. The average time cost of each training iteration is 0.25s on GPU, and it takes 1.4 hours for 20,000 iterations to complete the whole training process. Given a reasonably large mesh from the SCAPE dataset (12,500 vertices, 24,998 triangles) [12], it takes 26.4s to compute the raw SH descriptors from the mesh on CPU, 0.09s to compute the distance metric on GPU, 0.06s to compute the saliency map on GPU, and 163.4s to compute 30 saliency-induced embeddings from the metric on CPU. The method of [25] takes about 235.2s for shape matching on CPU.

A. Evaluation of Our Deep Learning Architecture

In this section, we validate that our RNN method is more effective at learning multi-scale shape features compared with the baselines in Fig. 7, and that our soft-thresholding operator further improves the performance via adaptive metric sparsification. We train on an 80% random sample of the 20 meshes from each of the 20 categories of the Schelling saliency dataset [3] and test on the remaining meshes at each training step. Here, we use only the saliency fitting loss for large-scale evaluation because none of the 20 categories apart from the Human and Fourleg has corresponding shape matching datasets [12]–[14]. We use the Gini index to measure the sparsity of saliency maps and metrics [57].

Evaluation of Our RNN. First, we evaluate our RNN method for multi-scale feature learning. To match our architecture, we stack 3 layers of MLPs with an input and an output dimension of 256 for each of the three baselines in Fig. 7, and use the tanh activation function and feature standardization for them. We find that the SGD parameters of our architecture work well for all of them as well. As shown in Fig. 9, neither a shared MLP nor multiple independent

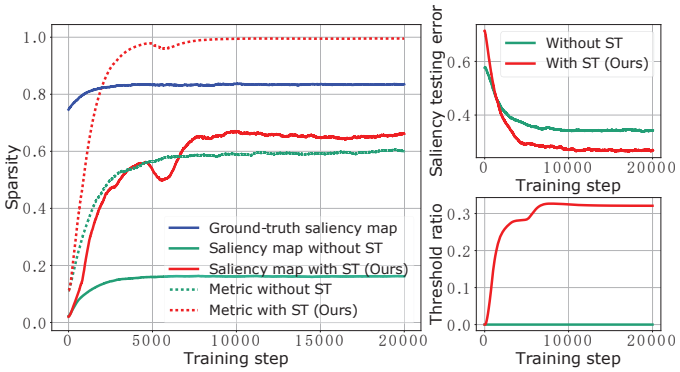


Fig. 10. **Soft-thresholding Evaluations**. Learning a threshold value (*bottom right*) to adaptively truncate the small elements of a metric to exact zeros considerably improves the sparsity of metric and drives the sparsity of saliency closer to that of the ground-truth (*left*). The resulting saliency testing error is also considerably lower (*top right*).

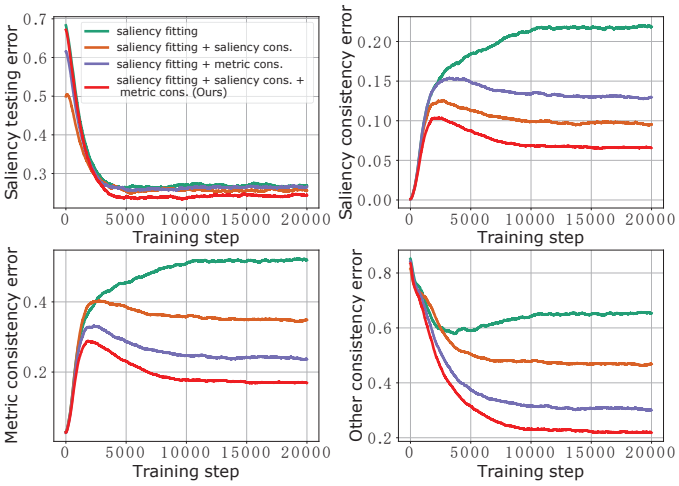


Fig. 11. **The Quantitative Evaluations of Saliency and Matching**. Learning with the saliency fitting loss, saliency consistency loss, and metric consistency loss together ($\alpha = 1, \beta = .02, \gamma = .02$) produces the lowest errors on all criteria, compared with when either the saliency or metric consistency losses are individually disabled, or when both are disabled. The *other consistency error* measures the difference of the metric without its principal eigenvector between two corresponding meshes.

MLPs perform well, because the former ignores the feature characteristics of different scales and the latter fail to integrate features across scales. A single MLP performs much better as it transforms the features of all scales simultaneously. Still, our RNN achieves the lowest saliency testing error by explicitly learning scale transition and integration, both with and without the soft-thresholding operator.

Evaluation of Our Soft-thresholding Operator. We then evaluate our soft-thresholding operator by training with and without it, as shown in Fig. 10. Without the soft-thresholding operator, although the ground-truth saliency maps gradually sparsifies the learned metric, the predicted saliency maps have relatively lower sparsity and considerable higher saliency testing error. Our operator improves the sparsification of the learned metric significantly by gradually learning a threshold to truncate small values, producing much better saliency maps with higher sparsity and lower testing error.

B. Mutual Benefits of Saliency and Matching

Here, we validate that jointly learning saliency and matching via our unified metric loss function enables each other to generalize better: while matching improves the accuracy and deformation-invariance of our computed saliency maps, saliency improves the semantic localization of our learned shape embeddings for more robust matching. We evaluate the saliency fitting loss, saliency consistency loss, and metric consistency loss all together, on the Human category of the Schelling saliency dataset [3] and the SCAPE matching dataset [12] (80% for training and 20% for testing). We perform another evaluation on the Fourleg category of the Schelling saliency dataset and the TOSCA matching dataset [13].

Quantitative Evaluations. As shown in Fig. 11, training with only the saliency fitting loss ($\alpha = 1$, $\beta = 0$, $\gamma = 0$) leads to the high saliency and metric consistency errors, indicating that neither the predicted saliency maps nor the metrics are sufficiently invariant to human body shape variations. Adding the saliency consistency loss alone ($\alpha = 1$, $\beta = .02$, $\gamma = 0$) improves the deformation-invariance of the predicted saliency maps a lot, but the metric remains sensitive to shape variations because only its principal eigenvector (i.e. saliency map) is regularized to be consistent. This can be seen from the high *other consistency error*, which measures the difference of the metric without its principal eigenvector between two corresponding meshes. Oppositely, adding the metric consistency loss alone ($\alpha = 1, \beta = 0, \gamma = .02$) leads to a more deformation-invariant metric by regularizing all eigenvectors together, but is less effective compared with the saliency consistency loss for inducing a deformation-invariant saliency map. In contrast, training with all three losses together ($\alpha = 1$, $\beta = .02$, $\gamma = .02$) produces the most deformation-invariant metrics and saliency maps, while achieving the lowest saliency testing error. The low metric consistency error along with the non-zero eigengaps (Fig. 5) ensures that our saliency-induced embeddings are deformation-invariant for shape matching.

Qualitative Evaluations. To compare the predicted saliency maps with and without matching, we train on a 5% (1 mesh), 10% (2 meshes), 20% (4 meshes), 40% (8 meshes), and 80% (16 meshes) sample of the respective dataset with and without the saliency and metric consistency losses. As shown in Fig. 12, under the extreme case of a single training mesh, the predicted saliency maps without consistency learning are full of unrecognizable noise. Remarkably, training with shape matching reduces the noise to a huge extent, allowing the identification of the salient regions of ears, hands, feet, and facial features. With more training meshes, the predicted saliency maps without consistency learning become less noisy, but they appear quite different between the two testing meshes, which suggests that they are sensitive to the non-rigid shape deformation. In contrast, the maps with consistency learning are much clearer and more consistent, even under the challenging settings of 2 and 4 training meshes. This confirms that overcoming intra-category shape variations via matching is the key to helping saliency detection generalize better, in both small and large sample training scenarios.

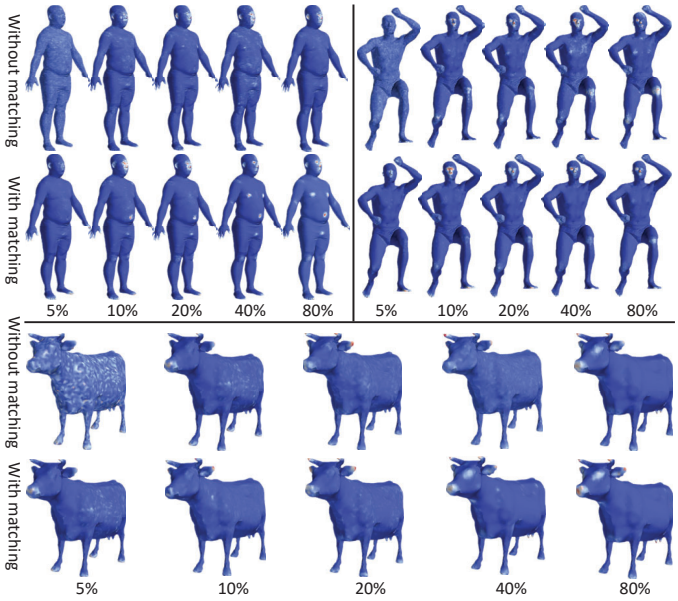


Fig. 12. **The Benefits from Matching to Saliency**. With matching, our computed saliency maps are less noisy and more sharply highlighted, especially in the extreme case of using one (5%) or two (10%) meshes for saliency training. The visual quality improvement of our saliency maps with matching is still noticeable with more meshes for saliency training.

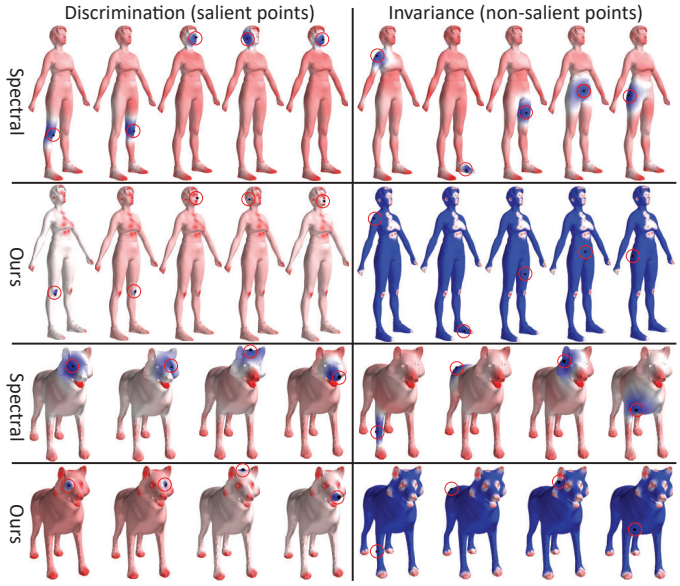


Fig. 13. **The Benefits from Saliency to Matching**. The red circle on each mesh highlights the reference point, and from there the distances to other points are represented using a blue (small) to red (large) scale. Using salient points as references (*left*), due to the semantic localization property, our saliency-induced embeddings discriminate these semantically important and thus deformation-stable points much better compared with the isometry-invariant spectral embeddings. Using non-salient points as references (*right*), we achieve maximum invariance for these points that are sensitive to non-isometric deformations.

We evaluate how saliency can help matching generalize better. We compare our embeddings with the Laplacian spectral embeddings because both of them are eigenvector solutions to the Laplacian embedding problem (2a) (2b) (2c) with salient affinity for the former and cotangent affinity for the later. As shown in Fig. 4, our embeddings are perfectly localized on the salient regions of ears, facial features, hands, and feet, while the spectral embeddings are globally supported on the mesh surface. Compared with the existing learned embeddings [26]–[30], ours are the first to achieve semantic localization while being guaranteed to be smooth and orthogonal as in the spectral embeddings. The semantic localization property would be difficult to obtain without the use of saliency that agrees with human annotations [3]. As shown in Fig. 13, our embeddings discriminate salient points more accurately (left), while maximizing the feature invariance among non-salient points since they are less reproducible under intra-category shape deformations (right). In contrast, the spectral embeddings provide an equally rough discrimination accuracy for each point on the shape, irrespective of whether it is salient or not. Our embeddings can therefore be used to prevent erroneous matchings from salient to non-salient points and vice versa, based on the consistency of saliency within a shape category [3].

C. Comparison with Saliency Detection Methods

Here, we compare our method with the highly-cited saliency detection methods, including mesh saliency (MS) [1], surface regions of interest (SRI) [15], manifold ranking (MR) [16], spectral irregularity (SI) [4], tree-based regression (TBR) [3], point neural networks (PointNet and PointNet++) [18], [20], and surface CNN (SurfCNN) [19]. Among them, MS is local contrast-based, SRI, MR, and SI are global rarity-based, and TBR is tree regression-based. Unlike PointNet that works on a raw 3D point cloud, PointNet++ and SurfCNN learn features using the geodesic metric and in the Laplacian spectral domain respectively. We input our raw SH features to PointNet++ and SurfCNN for a fair comparison. Note that our method does not use intrinsic geodesics or Laplacian but may incorporate them in the future.

Saliency Detection without Matching. As MS, SRI, MR, and SI are rule-based and cannot incorporate the intra-category consistency into saliency computation, we first train PointNet, PointNet++, and SurfCNN on a 80% sample (for each category) of the Schelling dataset and our method on a 5%, 10%, 20%, 40%, and 80% sample respectively. We find that our method produces the most accurate saliency maps using 80% training meshes.

Fig. 14 shows that MS responds strongly to local geometric variations while SRI, MR, and SI detect more globally distinct regions. As ground-truth saliency maps are spatially localized on surfaces, they must be densely distributed on the frequency dimension due to the well-known uncertainty principle. They are therefore not accurately captured by SI as it involves high-frequency cutoff in spectral computation. TBR produces good saliency maps with leave-one-out training, but fails to well reproduce the sparsity of ground-truth maps (e.g., on the face region of the human body shape). PointNet fails to identify most of ground-truth salient points, which are captured by PointNet++ and SurfCNN to some extent. However, PointNet++ and SurfCNN still misses some important regions such as the mouth and ears of the human body and the eyes of

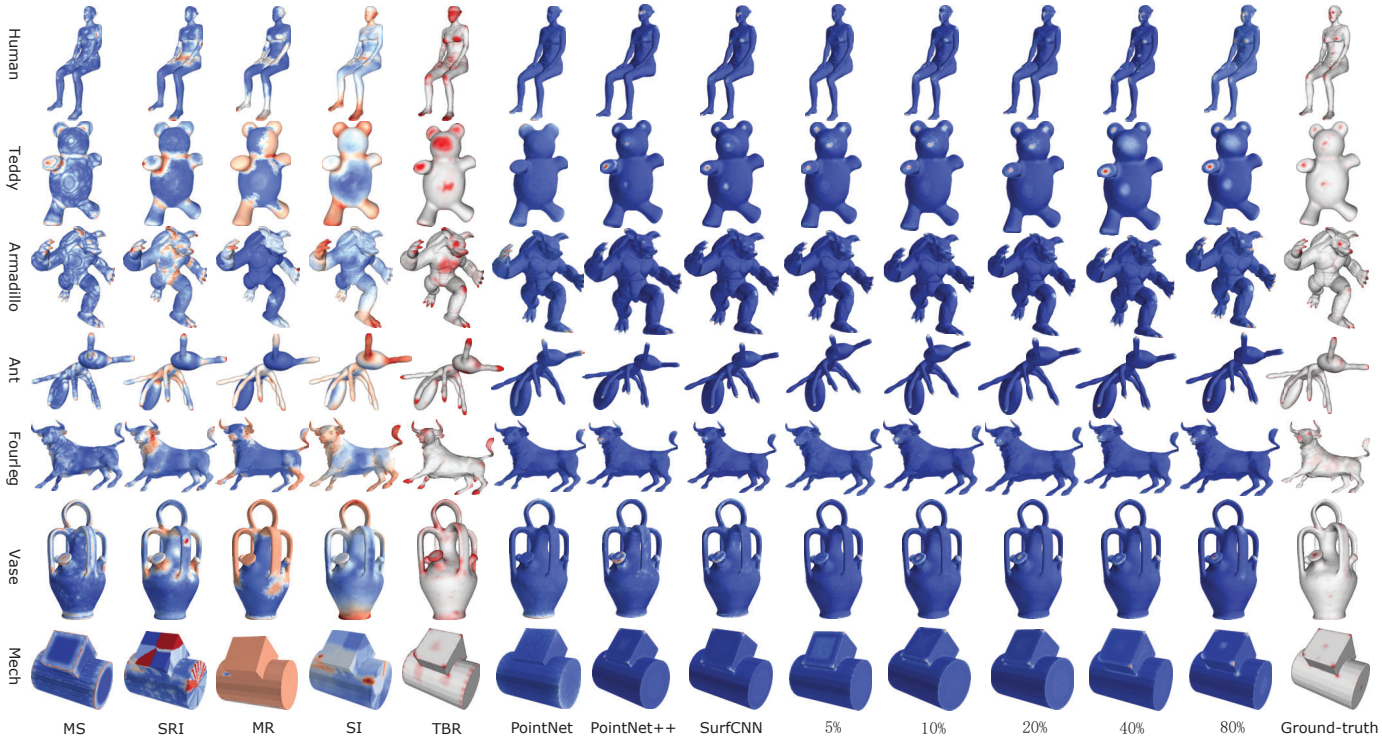


Fig. 14. **Visual Comparisons for Saliency Detection without Matching**. The image shows the saliency maps generated by MS, SRI, MR, SI, TBR, PointNet, PointNet++, SurfCNN, and our method, without the use of matching for saliency detection. Note that while PointNet, PointNet++ and SurfCNN are trained on a 80% sample for all the categories of the Schelling saliency dataset *jointly*, TBR is trained using leaving-one-out for each category *separately* in the original work. Our method is trained on varying fractions of samples for all the categories *jointly* to better visualize progression of generalization.

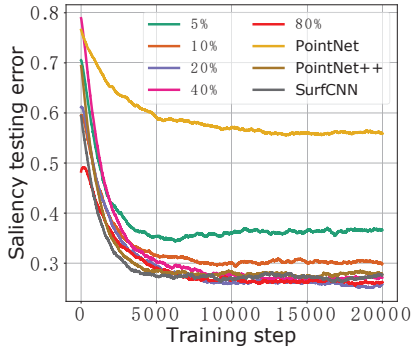


Fig. 15. **Quantitative Comparisons for Saliency Detection without Matching**. On average, the saliency maps predicted by our method with a 80% training sample are more accurate compared with that by PointNet, PointNet++, and SurfCNN with the same meshes for saliency training.

the cow. These regions are accurately captured by our method trained on an 80% sample. Even with as few as 5% or 10% training meshes, our method is shown to detect a succinct set of the most important regions such as the facial features and claws of the armadillo.

Fig. 15 shows that our method produces more accurate saliency maps than PointNet, PointNet++ and SurfCNN. It is interesting to see that our method achieves equally good quantitative results with a 20% and a 80% sample respectively, but adding more training meshes leads to visually smoother and more accurate saliency maps (Fig. 14).

Saliency Detection with Matching. We compare our method with PointNet, PointNet++, and SurfCNN by training with and without the saliency and metric consistency losses on the Human category of the Schelling saliency dataset [3] and the SCAPE matching dataset [12].

Fig. 16 shows the predicted saliency maps with and without matching for one testing mesh from the Schelling dataset on the left and another from the TOSCA dataset on the right. Incorporating matching into saliency detection reduces the noises on surfaces to a large extent, especially when there is only 1 training mesh providing no hints about the shape variations of testing meshes. When there are more training meshes, matching is shown to sharpen our detected salient regions such as the facial features of the human body on the left. PointNet fails to detect most of the salient regions, while PointNet++ does not highlight the facial features of the human body clearly. For the centaur shape on the right, PointNet++ and SurfCNN roughly capture the eyes, nose, and mouth of it with the help of matching. Our method highlights these regions more accurately when matching is used.

Fig. 17 shows that enforcing the intra-category consistency of saliency improves the saliency prediction accuracy of all methods except PointNet. The improvement is significant when there are only 1 (5%) or 2 (10%) training meshes but remains noticeable when there are more. Meanwhile, the considerably lower saliency consistency errors indicate that the predicted saliency maps are much more deformation-invariant with matching. Overall, our method achieves the lowest saliency prediction error using 80% of both saliency and matching training meshes.

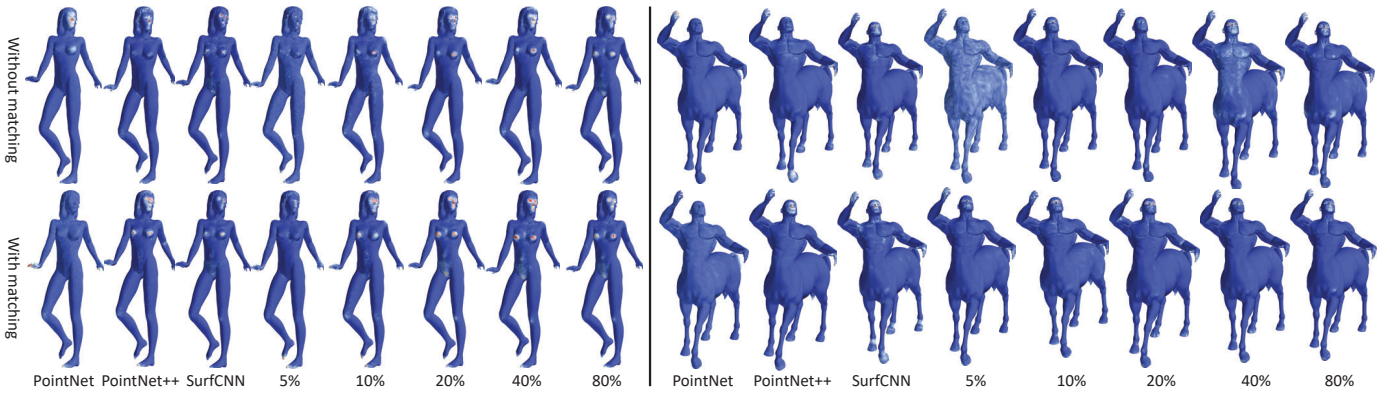


Fig. 16. **Visual Comparisons for Saliency Detection with Matching**. The image shows the saliency maps generated by PointNet, PointNet++, SurfCNN, and our method, with and without matching. PointNet, PointNet++, and SurfCNN are trained on a 80% sample of the Human category of the Schelling dataset and a 80% sample of the SCAPE dataset, and our method is trained in the same way but with varying fractions of meshes from the Schelling dataset.

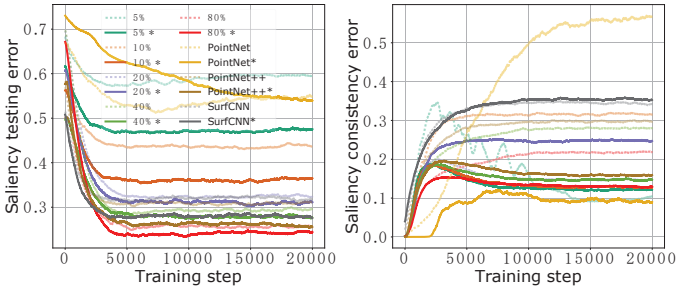


Fig. 17. **Quantitative Comparisons for Saliency Detection with Matching**. Compared with the saliency maps computed by PointNet, PointNet++, and SurfCNN, ours are more accurate (*left*) and deformation-invariant (*right*). We mark * to indicate the use of matching.

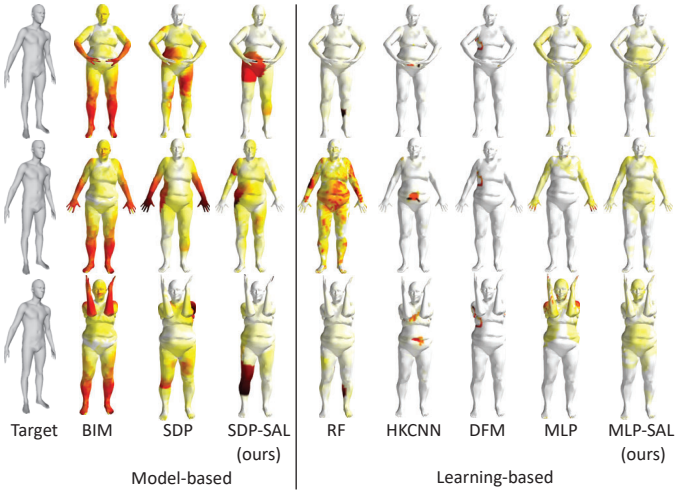


Fig. 18. **Visual Comparisons for Shape Matching with Saliency**. Visualization of the predicted correspondence error, i.e. geodesic distances between predicted and ground-truth correspondence points, from three source meshes to a target mesh on the FAUST testing set. Hotter colors indicate larger errors.

D. Comparison with Shape Matching Methods

Here, we compare our method with the highly-cited blended intrinsic maps (BIM) [7], semi-definite programming (SDP) [25], random forests (RF) [31], heat kernel CNN (HKCNN) [8], and deep functional maps (DFM) [33] for non-rigid shape matching. We group the methods into model-based and learning-based due to their different data requirements - the latter requires one-to-one vertex correspondences for training, while the former does not. We match each of the last 20 testing meshes to the first on the FAUST dataset for performance benchmarking, using the protocol of [7]. We obtain the correspondences by RF, HKCNN, and DFM for these meshes from the original authors, and run BIM and SDP for the same set of meshes using the released codes.

Saliency for Model-based Matching. We first incorporate our saliency-induced point embeddings (of dimensions 30, Fig. 4 bottom) into SDP, in addition to the originally used Laplacian spectral embeddings (of dimensions 30, Fig. 4 top), to better handle non-isometric shape deformations. We name our method of SDP with saliency as *SDP-SAL*. Fig. 18 shows some predicted correspondence error maps. It can be seen that BIM is inferior to SDP for matching the limbs of human bodies because it has no notion of length on surfaces. SDP produces patches of wrongly matched points due to its sensitivity to surface length-changing (non-isometric) deformations. Our SDP-SAL method reduces the matching errors of SDP at the limbs and chests using the saliency-induced embeddings. Fig. 19 shows that our SDP-SAL method achieves higher correspondence accuracy compared with SDP and BIM on the FAUST testing set. The consistent improvement from SDP to SDP-SAL indicates that saliency reduces the non-isometric correspondence errors that cannot be handled by the isometry-invariant spectral embeddings.

Saliency for Learning-based Matching. We then incorporate our saliency-induced embeddings into a three-layers plain MLP (of dimensions 256 for each layer) for correspondence prediction using our SH features on the FAUST training set (the first 80 meshes). We name our method of MLP with saliency as *MLP-SAL*. As RF and HKCNN refine the predicted correspondences using the functional maps of [6] and DFM uses the geodesic smoothing method of [58], we refine our MLP results using the method of [58] for a fair comparison. Our MLP-SAL method exploits both geodesic (as used by DFM) and our saliency-induced embedding distances (Fig. 13) for correspondence refinement. Fig. 18 shows that our MLP-SAL method reduces the matching errors produced by MLP

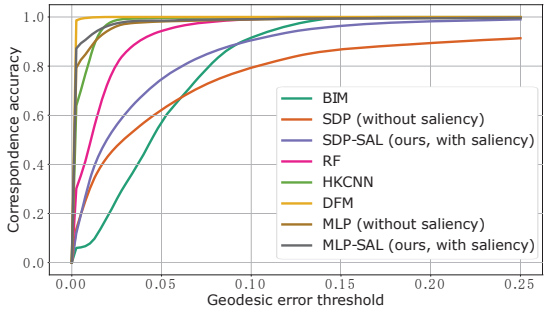


Fig. 19. **Quantitative Comparisons for Shape Matching with Saliency**. The comparison of the shape matching accuracy obtained by BIM, SDP (without saliency), RF, HKCNN, DFM, MLP (without saliency), as well as by our saliency-enhanced SDP-SAL and MLP-SAL on the FAUST testing set.

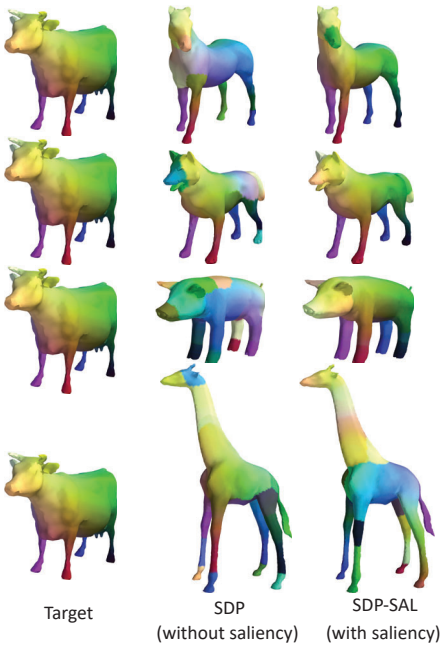


Fig. 20. **Matching Highly Non-Isometric Shapes with Saliency**. The image shows shape matchings generated by SDP and our SDP-SAL from four source meshes to a target mesh. These meshes are from the Fourleg category of the Schelling dataset, which is known to exhibit intra-category shape deformations that are far from being isometric.

at the shoulders and hands of human bodies. Fig. 19 shows that our MLP-SAL improves on MLP and achieves higher correspondence accuracy compared with RF and HKCNN.

More Non-Isometric Matching Results. To demonstrate the use of our saliency-induced embeddings for handling more complex intra-category shape variations, we compute shape matchings for the Fourleg category of the Schelling dataset using the isometry-invariant SDP and our saliency-enhanced SDP-SAL respectively. We extract our embeddings by training with an 80% sample of the Fourleg category and an 80% sample of the animal category of TOSCA shape matching dataset. Fig. 20 shows that these animal body shapes have strong non-isometric shape variations, which explains the failure of SDP to find semantically meaningful yet highly non-isometric shape matchings. Our SDP-SAL, in contrast, identifies correct matchings from the limbs of the horse, wolf, and pig to that of the cow. It also considerably reduces the matching errors of SDP at the face and back regions of the wolf and pig. For the even more challenging giraffe-to-cow example, only our SDP-SAL can identify correct matchings for the head region of the giraffe.

VII. Conclusion and Future Work

In this work, we tackled mesh saliency detection and non-rigid shape matching together for mutual benefits. We proposed a unified metric representation from which the saliency map and the shape embeddings of a mesh can be jointly inferred as the principal eigenvector and the smoothed Laplacian eigenvectors respectively. We also proposed a multi-layer RNN for effectively integrating multi-scale shape features, together with a soft-thresholding operator that adaptively enforces the sparsity of metric representation. We performed metric learning on saliency detection and shape matching datasets at the same time. Results validated that matching improves the accuracy and intra-category consistency of derived saliency maps, especially when the saliency training set is of small size (i.e. with only 1 or 2 meshes). They also showed that saliency improves the matching accuracy of both model-based and learning-based methods, which is more noticeable when large non-isometric deformations are involved.

Currently, our system requires dense point-to-point correspondences to enforce the intra-category consistency property, which have very limited availability and are difficult to label [12]–[14]. This may be partly addressed with sparse segment correspondences [6], but a more favorable bootstrap solution would be to compute less accurate matchings for improvement with target tasks jointly and iteratively.

The Laplacian spectral embeddings [23] and our saliency-induced ones represent the two extreme ends of discrimination-invariance tradeoff, with the former proven to be the smoothest and the latter proven to be the most localized. Therefore, our embeddings lack fine-grained discrimination for non-salient points. This is why we incorporate our embeddings into the model-based SDP and the learning-based MLP methods for shape matching. In between the Laplacian spectral embeddings and ours, there would be an optimal discrimination-invariance tradeoff that takes both salient and non-salient points into consideration. Finding the optimal solution depends on the applications and is a future direction.

The model-based SDP method [25] can handle asymmetric and bilaterally symmetric shapes (i.e. human and animal body shapes as shown in this paper), but it cannot easily handle more general symmetric cases because the convex solution set of the method strictly contains the non-convex solution set of the shape matching problem [59]. As a result, many solutions recovered by the method for general symmetric inputs do not correspond to a valid solution of shape matching. Fortunately, as proved in [59], the solution set of shape matching are actually the extreme points of the convex solution set of the method, from which a valid solution can be returned by maximizing random linear energies selected according to the uniform distribution on the unit sphere [60]. Adapting this method to handle general symmetric shapes will be our future work.

ACKNOWLEDGEMENTS

This work was supported in part by the Erasmus Mundus Action 2 Programme (Ref: 2014-0861/001-001), the Royal Society (Ref: IES\R2\181024), the National Key Research and Development Program of China (Ref: 2017YFB1002702), and the National Nature Science Foundation of China (Ref: 61572058). We would like to thank Kevin Mccay for polishing the paper.

REFERENCES

[1] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," *ACM Trans. on Graph.*, vol. 24, no. 3, pp. 659–666, 2005.

[2] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1681–1707, 2011.

[3] X. Chen, A. Saparov, B. Pang, and T. Funkhouser, "Schelling points on 3d surface meshes," *ACM Trans. on Graph.*, vol. 31, no. 4, p. 29, 2012.

[4] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, "Mesh saliency via spectral processing," *ACM Trans. on Graph.*, vol. 33, no. 1, p. 6, 2014.

[5] S.-W. Jeong and J.-Y. Sim, "Saliency detection for 3d surface geometry using semi-regular meshes," *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2692–2705, 2017.

[6] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Trans. on Graph.*, vol. 31, no. 4, p. 30, 2012.

[7] V. G. Kim, Y. Lipman, and T. Funkhouser, "Blended intrinsic maps," *ACM Trans. on Graph.*, vol. 30, no. 4, p. 79, 2011.

[8] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2016, pp. 3189–3197.

[9] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3196–3209, 2017.

[10] A. Toshev, J. Shi, and K. Daniilidis, "Image matching via saliency region correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[11] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT'2010*, 2010, pp. 177–186.

[12] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," *ACM Trans. on Graph.*, vol. 24, no. 3, pp. 408–416, 2005.

[13] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM Trans. on Graph.*, vol. 27, no. 3, p. 97, 2008.

[14] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3794–3801.

[15] G. Leifman, E. Shtrom, and A. Tal, "Surface regions of interest for viewpoint selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 414–421.

[16] P. Tao, J. Cao, S. Li, X. Liu, and L. Liu, "Mesh saliency via ranking unsalient patches in a descriptor space," *Computers and Graphics*, vol. 46, pp. 264–274, 2015.

[17] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3d shape surfaces using geometry images," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 223–240.

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] L. Yi, H. Su, X. Guo, and L. Guibas, "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2017, pp. 5099–5108.

[21] H. Xiao, J. Feng, Y. Wei, M. Zhang, and S. Yan, "Deep salient object detection with dense connections and distraction diagnosis," *IEEE Transactions on Multimedia*, 2018.

[22] K. Fu, I. Y.-H. Gu, and J. Yang, "Saliency detection by fully learning a continuous conditional random field," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1531–1544, 2017.

[23] R. M. Rustamov, "Laplace-beltrami eigenfunctions for deformation invariant shape representation," in *The fifth Eurographics Symposium on Geometry Processing*, 2007, pp. 225–233.

[24] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[25] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman, "Point registration via efficient convex relaxation," *ACM Trans. on Graph.*, vol. 35, no. 4, p. 73, 2016.

[26] É. Corman, M. Ovsjanikov, and A. Chambolle, "Supervised descriptor learning for non-rigid shape matching." in *ECCV Workshops*, 2014, pp. 283–298.

[27] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *TPAMI*, vol. 36, no. 1, pp. 171–180, 2014.

[28] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers, "Anisotropic diffusion descriptors," *Computer Graphics Forum*, vol. 35, no. 2, pp. 431–441, 2016.

[29] L. Cosmo, E. Rodola, J. Masci, A. Torsello, and M. M. Bronstein, "Matching deformable objects in clutter," in *3D Vision*, 2016, pp. 1–10.

[30] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1544–1553.

[31] E. Rodolà, S. Rota Bulo, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4177–4184.

[32] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Computer Graphics Forum*, vol. 34, no. 5, pp. 13–23, 2015.

[33] O. Litany, T. Remez, E. Rodola, A. M. Bronstein, and M. M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *Proceedings of the International Conference on Computer Vision*, vol. 2, 2017, p. 8.

[34] A. B. Hamza and H. Krim, "Geodesic matching of triangulated surfaces," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2249–2258, 2006.

[35] S. Biasotti, A. Cerri, M. Aono, A. B. Hamza, V. Garro, A. Giachetti, D. Giorgi, A. Godil, C. Li, C. Sanada *et al.*, "Retrieval and classification methods for textured 3d models: a comparative study," *The Visual Computer*, vol. 32, no. 2, pp. 217–241, 2016.

[36] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3d shape retrieval," *Neurocomputing*, vol. 204, pp. 41–50, 2016.

[37] M. Reuter, F.-E. Wolter, and N. Peinecke, "Laplace–beltrami spectra as 'shape-dna'of surfaces and solids," *Computer-Aided Design*, vol. 38, no. 4, pp. 342–366, 2006.

[38] S. Biasotti, A. Cerri, M. Abdelrahman, M. Aono, A. B. Hamza, M. El-Melegy, A. Farag, V. Garro, A. Giachetti, D. Giorgi *et al.*, "Shrec'14 track: Retrieval and classification on textured 3d models," in *Proceedings of the Eurographics workshop on 3d object retrieval*, 2014, pp. 111–120.

[39] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Proceedings of Shape Modeling Applications*, 2004, pp. 167–178.

[40] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong, "3d deep shape descriptor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2319–2328.

[41] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[42] M. Savva, F. Yu, H. Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, H. Su, S. Bai, X. Bai *et al.*, "Shrec16 track: largescale 3d shape retrieval from shapenet core55," in *Proceedings of the Eurographics Workshop on 3D Object Retrieval*, 2016, pp. 89–98.

[43] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.

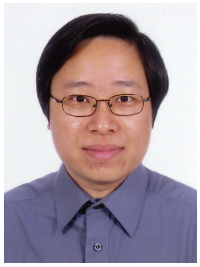
[44] J. Harel, C. Koch, P. Perona *et al.*, "Graph-based visual saliency," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2006, p. 5.

[45] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2002, pp. 585–591.
[46] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
[47] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3d shape descriptors," in *Eurographics Symposium on Geometry Processing*, vol. 6, 2003, pp. 156–164.
[48] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 448–456.
[49] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *arXiv preprint arXiv:1808.03314*, 2018.
[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[51] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv:1409.1259*, 2014.
[52] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12–30, 2015.
[53] X.-T. Yuan and T. Zhang, "Truncated power method for sparse eigenvalue problems," *J. Mach. Learn. Res.*, vol. 14, no. 4, pp. 899–925, 2013.
[54] J. R. Magnus, "On differentiating eigenvalues and eigenvectors," *Econometric Theory*, vol. 1, no. 02, pp. 179–191, 1985.
[55] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," *IJCV*, vol. 96, no. 1, pp. 28–45, 2012.
[56] E. Corman, J. Solomon, M. Ben-Chen, L. Guibas, and M. Ovsjanikov, "Functional characterization of intrinsic and extrinsic geometry," *ACM Trans. on Graph.*, vol. 36, no. 2, p. 14, 2017.
[57] N. Hurley and S. Rickard, "Comparing measures of sparsity," *IEEE Trans. on Information Theory*, vol. 55, no. 10, pp. 4723–4741, 2009.
[58] M. Vestner, R. Litman, E. Rodolà, A. Bronstein, and D. Cremers, "Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[59] N. Dym and Y. Lipman, "Exact recovery with symmetries for procrustes matching," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1513–1530, 2017.
[60] A. I. Barvinok, "Problems of distance geometry and convex properties of quadratic maps," *Discrete & Computational Geometry*, vol. 13, no. 2, pp. 189–202, 1995.

Shanfeng Hu is a final-year PhD student from the Department of Computer and Information Sciences at Northumbria University, UK. His recently submitted thesis is about incorporating structured metric representations for 3D geometric deep learning. He also works part-time as a Senior Data Scientist at tenokonda Ltd. on deep learning for quantitative data analytics. He received his Bachelor degree from the Department of Computer Science and Technology at Henan University, China.

Hubert P. H. Shum is an Associate Professor in Computer Science at Northumbria University, UK, and the Director of Research and Innovation of the Computer and Information Sciences Department. Before that, he was a Senior Lecturer at Northumbria University, a Lecturer at the University of Worcester and a Post-doctoral Researcher at RIKEN Japan. He received his PhD degree from the University of Edinburgh, his Master and Bachelor degrees from the City University of Hong Kong. He serves as an Associate Editor in Computer Graphics Forum.

Nauman Aslam is an Associate Professor in the Department of Computer Science and Digital Technologies, Northumbria University, UK. He is also an Adjunct Assistant Professor at Dalhousie University, Canada. He received his PhD in Engineering Mathematics from Dalhousie University, Halifax, Nova Scotia, Canada in 2008. Prior to joining Northumbria University, he worked as an Assistant Professor at Dalhousie University, Canada. His research interests include wireless sensor network, energy efficiency, security and WSN health applications.



Frederick W. B. Li received a B.A. and an M.Phil. degree from Hong Kong Polytechnic University, and a Ph.D. degree from the City University of Hong Kong. He is currently an Assistant Professor at Durham University. He served as a guest Editor for several special issues of World Wide Web Journal, Journal of Multimedia and JDET. He has served as a Program Co-Chair of ICWL for four years and IDET for two years. His research interests include distributed virtual environments, computer graphics and e-Learning systems.

Xiaohui Liang received his Ph.D. degrees in computer science and engineering from Beihang University, China. He is currently a Professor, working at the School of Computer Science and Engineering at Beihang University. His main research interests include computer graphics and animation, visualization and virtual reality.